REMARKS

Claims 1, 4-11, and 14-31 remain in the application. Claim 2 was cancelled in a previous amendment. Claims 3, 4, 12, and 13 are now canceled. Claims 1, 11, 14, 18, 20 and 24 have been amended. Applicant respectfully requests reconsideration of the pending claims in light of the amendments and the following remarks.

CLAIM REJECTIONS UNDER 35 USC §112

Claims 1 and 3-31 were rejected under 35 USC 112, first paragraph, as failing to comply with the written description requirement. The amendments moot this rejection. Applicant respectfully disagrees with the Examiner that the independent claims call for an intentional determination to distinguish whether the system runs in a distributed memory system. The written description clearly disclosed two embodiments: (1) system implemented in a distributed memory system and (2) system implemented in a shared memory system. Note that FIGs. 3 and 4 show a shared memory machine and a distributed memory machine.

CLAIM REJECTIONS UNDER 35 USC §103

The Office Action rejected claims 1 and 3-31 under 35 USC 103(a) as being unpatentable over Long et al. (US 2002/0129079 A1) (hereinafter "Long") in further view of Tanenbaum et al. [Distributed Systems: Principles and Paradigms] (hereinafter "Tanenbaum") and Tremblay (US 2001/0042188) (hereinafter Tremblay).

Claim 1 has been amended to expressly claim a plurality of processors and a global

8

address space language program that has a plurality of program threads that access memory in a global address space system. The combination of references cited in the Office Action neither teaches nor suggests these limitations. There is no mention or hint in the cited prior art of a global address space language program that has a plurality of program threads that access memory in a global address space system.

As to claims 3, 13, 26 and 27, the Examiner conceded that Long does not teach or suggest implementing the runtime system on a distributed memory system; and the "directory of shared variables stored in a private memory of each thread such that the directory is replicated across all of the threads." The non-final Office Action states that "Tanenbaum teaches that a distributed memory system (see Tanenbaum, page 16, Fig. 1-6, Private Memory), provides fault-tolerance and increased storage and processing capabilities of the processing system." Here Tanenbaum is merely describing the advantages of a distributed memory system and does not teach or suggest the directory of shared variables stored in the private memory of all threads.

The non-final Office Action further states that: "Tanenbaum further teaches that full-replication of shared data provides further fault-tolerance." Again, Tanenbaum is describing a benefit of full replication so that operations may be performed on shared data if one element is lost. Tanenbaum neither teaches nor discloses the claimed subject matter of *how* that shared data is replicated. There are different methods of replicating data and Tanenbaum does not teach the method as claimed in the instant application. Tanenbaum touches on the pros and cons of replication using the example of replication of web pages by caching. He states that an advantage of replication is faster access time, but a disadvantage of replication is "that more

network bandwidth is now consumed keeping all replicas up-to-date." [Tanenbaum pg. 293, lines 6-8] Here Tanenbaum is actually teaching away from the claimed subject matter wherein "A control structure is used to control access to the shared data such that all threads can access the data at any time. Since all threads see the same objects, synchronization issues are eliminated. In addition, the improved efficiency of the data sharing allows the number of program threads to be vastly increased." [Abstract]

There is absolutely no teaching or suggestion in Tanenbaum to store a directory of shared variables in a private memory of each thread across a distributed or shared system such that the directory is replicated across all of the threads. On the subject of replication, Tanenbaum merely provides a textbook recitation that full-replication of shared data may be beneficial.

Long does not teach or suggest providing all threads in a system with global and synchronized access to shared data so that all threads in the system can "see" the same data. Long teaches tracking monitors used for acquiring access to objects in order to determine if the monitor is suitable for reclamation during a garbage collection process. Long discloses the use of locks and lock counts for tracking so that multiple threads cannot access the same object concurrently. Nowhere does Long teach or suggest the elements of claim 1; in fact, Long teaches away from the claimed subject matter with his dependence on locks and lock counts for restricting concurrent access. Tremblay does not teach the claimed subject matter of a global address space language program.

Claims 5 – 10 are dependent upon claim 1; therefore they are patentable for at least the same reasons that their parent claim, claim 1, is patentable.

Claim 11 is directed to the shared memory machine embodiment of the invention. It also requires a global address space language program that has a plurality of program threads that access memory in a global address space system. None of the three references on which the rejection is based even hints at this claim element.

Claim 12 is dependent upon claim 11; therefore claim 12 is patentable for at least the same reasons that its parent claim is patentable.

Claims 13, 14, and 15 are dependent upon claim 11; therefore they are patentable for at least the same reasons that their parent claim is patentable.

Claims 16 – 19 are dependent on claim 15 which is dependent on claim 11; therefore they are patentable for at least the same reasons that claim 11 is patentable.

Claim 20 – 23 are dependent upon claim 11; therefore they are patentable for at least the same reasons that their parent claim is patentable.

Claim 24 has been amended to require running a global address space language program comprising a plurality of program threads. It is therefore patentable over the combination of references for the reason given above.

Claim 25 is dependent on claim 24; therefore it is patentable for at least the same reasons that claim 24 is patentable.

Claim 28 is dependent on claim 26 which is dependent on claim 24; therefore claim 28 is patentable for at least the same reasons that claim 24 is patentable.
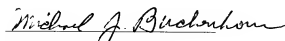
Claims 29 and 30 are dependent on claim 24; therefore they are patentable for at least the same reasons that claim 24 is patentable.

Claim 31 is dependent on 29, which is dependent on claim 24; therefore it is patentable for at least the same reasons that claim 24 is patentable.

For the foregoing reasons, Applicant respectfully requests allowance of the pending claims.

Respectfully submitted,

Michael J. Buchenhorner
Reg. No. 33,162

E-filed on Date: June 4, 2007

Michael Buchenhorner, P.A.
8540 S.W. 83 Street
Miami, Florida 33143
(305) 273-8007 (voice)
(305) 595-9579 (fax)